with the database kernel include an entity relation manager for modifying the linker table to dynamically define entity relationships among the information entities; a hierarchal structure manager for dynamically defining hierarchal relationships among objects in the information entities; a load/unload utility, for changing information between a human-readable format and another format having a syntax acceptable to the database kernel; and an editor which provides an environment that enables a human operator to conveniently interact with the relational database.

INDEPENDENT CLAIMS

1. In an object persistence management system, a many-to-many relationship manager comprising:

a plurality of related objects;

a junction table storing relationships between said related objects; and,

a plurality of corresponding links, each said link corresponding to one of said objects, each said link persisting state information for said corresponding object in an associated object table, and managing said junction table responsive to changing relationships with others of said related objects.

5. A method of managing a many-to-many relationship in an object persistence management system comprising the steps of:

detecting a relationship change with a related object;

storing a directive in a buffer, said directive specifying a management operation for changing said relationship in a junction table; and,

performing said stored directive only if an opposite directive has not been stored in a buffer associated with said related object.

8. A machine readable storage having stored thereon a computer program for managing a many-to-many relationship in an object persistence management system, the computer program comprising a routine set of instructions for causing the machine to perform the steps of:

detecting a relationship change with related object;

storing a directive in a buffer, said directive specifying a management operation for changing said relationship in a junction table; and,

performing said stored directive only if an opposite directive has not been stored in a buffer associated with said related object.

SUBJECT FOR DISCUSSION UNDER M.P.E.P. § 713.09

1. Would the Examiner agree that claim 1 recites the limitation, "each said link ... managing said junction table responsive to changing relationships with others of said related objects."?

2. Would the Examiner agree that the following text is the entirety of the cited portion of column 6, lines 37 through 45?

> Another aspect of the invention is the capability of dynamically defining many-to-many relationships between information entities, and further of identifying individual objects within each information entity which are specifically related. The defined many-to-many relationships are maintained in a single table (the "CZ" table) which stores information instantiating relationships between specific objects in separate tables. In this respect the invention differs from the known prior art, which requires multiple linker tables to hold a plurality of many-to-many entity relationships. As defined in the SQL file shown in Appendix "E" at E1 (the "CA" table), the columns of the linker table contain: the relationship identification number (entity relation number) of the instantiation; the unique identification of object no. 1; the unique identification of object no. 2; parametric identifiers, if applicable; and a descriptor for the relationship between the objects.

A linker table in accordance with the definitional information of the CA table is shown at E2 of Appendix "E" (the "CZ" table). The table at E3 of Appendix "E" presents sample rows from the CZ table in user-readable format. The user can normalize the database and maintain the linker table by invoking entity relation manager 32 in cooperation with editor 26, forms manager 28, and load/unload utility 30. The CZ table is accessed by kernel 24 when information is retrieved from the database, for example by a select statement. Although the contents of the CZ table are modified when many-to-many relationships are defined, the structural organization of the tables in the database is completely undisturbed.

3. Would the Examiner agree that nowhere in the foregoing citation of Olson is it ever suggested that a "link" manages a junction table?

4. Would an amendment indicating inserting the language "each said link" before "managing said junction table" suffice to clarify that the links manage the junction table?

4. Would the Examiner point out with specificity where in Olson the Applicants can find the teaching recited in claims 5 and 8 as follows: "performing said stored directive only if an opposite directive has not been stored in a buffer associated with said related object."

CONCLUSION

The Applicant through undersigned counsel eagerly anticipates the personal interview of August 18, 2005 and, as such, the Applicant requests that the Examiner call the undersigned if clarification is needed on any matter within this proposed agenda.

Respectfully submitted,

Date: August 15, 2005

Steven M. Greenberg, Reg. No.: 44,725
Attorney for Applicant(s)
Christopher & Weisberg, P.A.
200 East Las Olas Boulevard, Suite 2040
Fort Lauderdale, Florida 33301
Customer No. 31292